

Serializácia dát medzi klientom a API rozhraniami

- Serializácia typu Unival
- Implicitná konverzia jednoduchých JSON typov na Unival hodnotu
- Definovanie vrátených hodnôt z RPC metód
- Optimalizácia obsahu vrátenej hodnoty Unival
- Unival typu štruktúra ("record")
- Transformácie asových radov
 - Downsampling
 - OHLC

Obe rozhrania Comet aj REST API zdieajú ten istý spôsob serializácie komunikácie prostredníctvom [JSON formátu](#). Výnimku tvorí len volanie SBA RPC metód, ktoré slúžia na posielanie a získavanie binárnych dát z D2000 - v tomto prípade sa dáta posielajú v binárnej forme. JSON formát je univerzálnym formátom implementovaným sná v každom programovacím jazyku. Jeho výhodou je nielen natívna podpora v každom prehliadači ale aj jednoduchá čitateľnosť ľuďmi, pretože sa jedná o textový formát. Nevýhoda väčšieho objemu správy oproti akémukoľvek binárnemu formátu je minimalizovaná použitím gzip kompresie implicitne podporovanej ako v REST tak aj v Comet API.

Serializácia typu Unival

Základnou jednotkou výmeny dát medzi klientom aj D2000 systémom je [typ Unival](#) - zoskupujúci [základné atribúty objektov v D2000](#). Nasledujúci príklad unival hodnoty v JSON formáte, reprezentuje hodnotu typu reálne číslo so [stavom Valid](#).

Príklad zápisu Unival hodnoty

```
{
  "type": "real",
  "value": 123.456,
  "status": ["Valid"]
}
```

Každý Unival hodnota má jasne definovaný typ pomocou atribútu type. V prípade posielania Unival hodnôt do D2000 systému atribút type musí byť vždy definovaný. Výnimku tvorí iba možnosť ke namiesto objektového zápisu Unival typu, pošleme priamo hodnotu, v našom prípade 123,456. Tá je SmartWeb serverom automaticky konvertovaná na typ "real", keďže sa jedná o hodnotu s desatinnou čiarkou (v prípade poslania reazca by bol typ nastavený automaticky na hodnotu "text" a celého čísla na hodnotu "int").

Skrátený zápis Unival hodnoty

123.456

Odpoveď zo SmartWeb servera, nikdy nechodí v tomto skrátenom zápise, ale v "objektovom" zápise s definovaným typom (atribút type) a hodnotou (atribút value) ak je platná.

Ostatné atribúty nie sú v odpovedi zo SmartWeb servera implicitne vrátené kvôli optimalizácii, ale klient si ich vie vyžiadať cez špeciálny atribút returnAs (popísaný nižšie).

Atribút type môže nadobúdať nasledovné hodnoty:

Hodnota atribútu type	Popis typu hodnoty
"nan"	žiadna
"bool"	typ <i>VBool</i>
"int"	celé číslo
"real"	reálne číslo
"station"	typ <i>VStation</i>
"alarm"	typ <i>VAlarm</i>
"process"	typ <i>VProcess</i>
"time"	celé číslo (počet milisekúnd od epochy)
"timespan"	reálne číslo (počet sekúnd)

"text"	text
"array"	pole objektov <i>D2ApiValue</i>
"qval"	typ <i>VQval</i>
"record"	dvojmerné pole (riadok, stpec) hodnot

Zoznam všetkých atribútov Unival objektu je vypísaný v nasledujúcej tabuľke:

Atribút	Typ hodnoty	Povinný	Predvolená hodnota	Poznámka
type	text	áno		
value	poda typu	nie		nenastavený atribút automaticky znamená neplatnú hodnotu, nastavený platnú (ak príznak nie je preažený v atribúte status)
valueTime	celé íslo (poet milisekúnd od epochy)	nie	aktuálny as	asová znaka
valueTimes	dvojmerné pole (riadok, stpec) celých ísel (poet milisekúnd od epochy)	nie	aktuálny as	asové znaky hodnot v štruktúre, len pre typ "record"
alarmTime	celé íslo (poet milisekúnd od epochy)	nie		asová znaka alarmu
alarmTimes	dvojmerné pole (riadok, stpec) celých ísel (poet milisekúnd od epochy)	nie		asové znaky alarmov v štruktúre, len pre typ "record"
flags	pole textov (vymenovaný typ Flag)	nie	žiadny príznak	pole užívateľských príznakov, možné hodnoty sú "A" až "P"
flagsSets	dvojmerné pole (riadok, stpec) polí textov (vymenovaný typ Flag)	nie	žiadny príznak	dvojmerné pole polí užívateľských príznakov hodnot štruktúry, len pre typ "record"
limitStatus	text (vymenovaný typ LimitStatus)	nie	"InLimit"	limitný stav, možné hodnoty sú: "InLimit", "VeryLow", "Low", "High", "VeryHigh", "LimitsProblem"
limitStatuses	dvojmerné pole (riadok, stpec) polí textov (vymenovaný typ LimitStatus)	nie	"InLimit"	dvojmerné pole limitných stavov hodnot štruktúry, len pre typ "record"
processAlarmStatus	texto (vymenovaný typ ProcessAlarmStatus)	nie	"NoAlarm"	stav alarmu procesu, možné hodnoty sú: "NoAlarm", "ToOn", "ToOff", "On", "Off", "Err", "Oscillate", "ErrCmdOn", "ErrCmdOff", "SwToTrans", "SwToOff", "SwToOn", "SwToErr", "SwTrans", "SwOff", "SwOn", "SwErr", "ErrZalCmdOff", "HL", "VHL", "LL", "VLL", "ToHL", "ToVHL", "ToLL", "ToVLL", "ErrWriteCmd", "Change", "A29", "A30", "A31", "SysPrAl"
processAlarmStatuses	dvojmerné pole (riadok, stpec) polí textov (vymenovaný typ ProcessAlarmStatus)	nie	"NoAlarm"	dvojmerné pole stavov alarmu procesov v štruktúre, len pre typ "record"
status	pole textov (vymenovaný typ Status)	nie	príznak "Valid"	pole stavov, možné hodnoty sú: "Valid", "ProcAlarm", "NoAckPAlarm", "PrAlSilent", "Weak", "NoAckValue", "Transient", "Default", "Manual", "AlCrit", "Unknown"
statusSets	dvojmerné pole (riadok, stpec) polí textov (vymenovaný typ Status)	nie	príznak "Valid"	dvojmerné pole stavov hodnot štruktúry, len pre typ "record"
formattedValue	text	nie		v atribúte sa vracia v odpovedi zo servera formátovaná hodnota objektu D2000, nemá význam pri volaniach RPC metód
structType	text	áno		meno štruktúry, len pre typ "record", povinný na každom type "record" posielanom do D2000
definition	objekt <i>D2RecordDefinition</i>	-		definícia štruktúry, len pre typ "record", nastavený na každej hodnote s typom "record" vracanej z D2000
returnAs	text	nie		má význam len pri volaní RPC metód s výstupnými parametrami, definuje logický názov, pod ktorým bude vrátená výstupná hodnota

returnFields	pole textov	nie	prázdné pole	špeciálny atribút definuje dodatočné návratové atribúty požadované klientom od servera
returnTransformation	objekt <i>ReturnTransformation</i>	nie	null	len pre typ "record" s jedným íselným stĺpcom a rastúcimi asovými znakami hodnôt, obsahuje konfiguráciu spracovania (downscalingu) asového radu, vykonávanom na serveri pred posielaním odpovede klientovi, kvôli veľkému rozsahu dát, hodnoty sú popísané nižšie

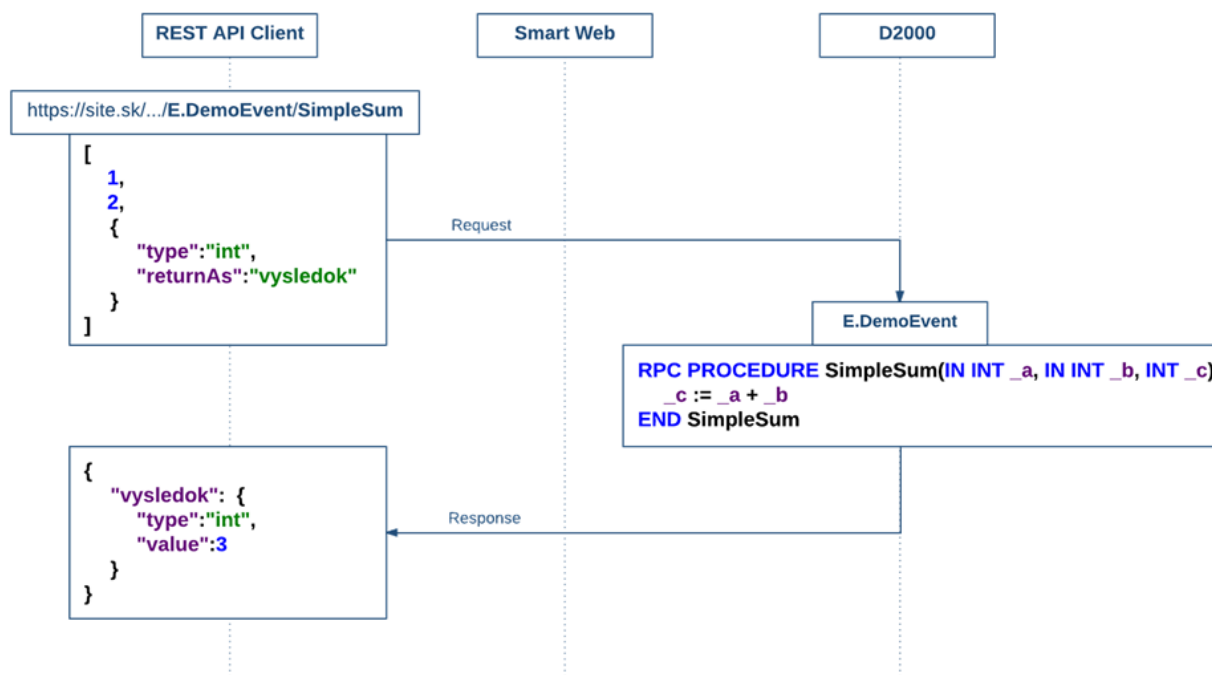
Implicitná konverzia jednoduchých JSON typov na Unival hodnotu

Pre zjednotenie práce cez API rozhrania je možné ako vstupné hodnoty do systému D2000 používať aj jednoduché typy namiesto objektov typu Unival. Takéto hodnoty majú automaticky nastavený príznak platnosti, as vzniku na aktuálny as a ostatné príznaky nadobúdajú ich predvolenú hodnotu.

Typ formátu JSON	Zodpovedajúci typ Unival	Poznámka
boolean	"bool"	
number	"bool"	Ak je cieový typ v D2000 <i>BOOL</i> . Ísla sú interpretované podľa poradia v type <i>VBool</i> .
number	"int"	Ak je cieový typ v D2000 <i>INT</i> . Ak bolo zadané reálne číslo, tak bude zaokrúhlené na celé.
number	"real"	Ak je cieový typ v D2000 <i>REAL</i> .
number	"time"	Ak je cieový typ v D2000 <i>TIME</i> . Pozor: číslo je interpretované ako počet sekúnd od 1972-01-01 00:00:00 UTC, nie ako počet milisekúnd od epochy.
string	"text"	

Definovanie vrátených hodnôt z RPC metód

Atribút returnFields definuje logický názov výstupného parametra požadovaného klientom od servera.



Na obrázku je znázornený príklad volania metódy `SimpleSum` klientom. Prvé dva parametre sú jednoduché JSON typy ktoré sú implicitne konvertované na správne Unival hodnoty. Tretí parameter je výstupný a klient definuje v atribúte `returnAs`, logický názov pod ktorým mu bude výstupná unival hodnota vrátená, v tomto prípade je to logický názov `"vysledok"`.

Optimalizácia obsahu vrátenej hodnoty Unival

Kvôli minimalizácii prenášaných dát Unival objekty štandardne na výstupe z D2000 obsahujú len atribút `type` a `value` (ak je hodnota platná). V prípade potreby ostatných rozširujúcich atribútov hodnoty, ktoré poskytuje systém D2000, je možné pri volaní nastaviť atribút `returnFields` a v ňom vymenovať požadované atribúty. Hodnota atribútu `returnFields` má formát poľa textov. Prípustné hodnoty sú v nasledujúcej tabuľke.

Hodnota v poli returnFields	Vypané atribúty v odpovedi zo servera
"AlarmTime"	alarmTime, alarmTimes
"Flags"	flags, flagsSets
"LimitStatus"	limitStatus, limitStatuses
"ProcessAlarmStatus"	processAlarmStatus, processAlarmStatuses
"Status"	status, statusSets
"ValueTime"	valueTime, valueTimes
"FormattedValue"	formattedValue

Unival typu štruktúra ("record")

Hodnota typu štruktúra má svoje hodnoty a atribúty uložené ako dvojrozmerné pole (riadok, stpec). Hodnoty nadobúrajú typ podľa definície štruktúry v systéme D2000. Nepoužíva rozširujúce atribúty alarmTime, flags, limitStatus, processAlarmStatus, status a valueTime. Namiesto nich používa atribúty alarmTimes, flagsSets, limitStatuses, processAlarmStatuses, statusSets, valueTimes, ktoré sú dvojrozmerným poom a prvky majú rovnaký typ ako pôvodné atribúty. alšími atribútmi používanými len pri štruktúrach sú definition a structType. Atribút definition (objekt *D2RecordDefinition*) je vždy automaticky nastavený na všetkých štruktúrovaných hodnotách, ktoré vystupujú zo systému D2000. Tento atribút popisuje názvy stpcov a ich typy v štruktúrovanej premennej. Typy stpcov v štruktúre sú zjednodušenou verziou typov Unival.

Typ stpca	Typ hodnoty
"bool"	typ <i>VBool</i>
"integer"	celé číslo
"real"	reálne číslo
"time"	celé číslo (poet milisekúnd od epochy)
"timespan"	reálne číslo (poet sekúnd)
"text"	text
"object"	podľa typu referencovaného D2000 objektu

Atribút structType je textový a uruje meno objektu D2000 typu definícia štruktúry. Je povinné ho nastaviť pre každú štruktúrovanú hodnotu, ktorá vstupuje do systému D2000.

Príklad posielanej hodnoty typu štruktúra - definícia SD.ArrReal_Text, 2 stpce (int, text), 3 riadky:

```
{
  "type": "record",
  "structType": "SD.Arr_Real_Text",
  "value": [[1, "One"], [2, "Two"], [3, "Three"]]
}
```

Príklad vracanej hodnoty typu štruktúra - definícia SD.ArrReal_Text, 2 stpce (int, text), 3 riadky:

```
{
  "type": "record",
  "definition": {
    "columnTypes": ["integer", "text"],
    "columnNames": ["digit", "name"]
  },
  "value": [[1, "One"], [2, "Two"], [3, "Three"]]
}
```

Transformácie asových radov

Pod asovým radom sa rozumie jednotpcová štruktúra, kde hodnoty sú reálne ísla a majú nastavené rastúce asové znaky. Pri zobrazení asových radov aso nie sú potrebné alebo žiadané (najmä kvôli výkonu) všetky hodnoty z požadovaného intervalu. Pre takéto potreby sú implementované transformácie asových radov, ktoré zmenšujú počet prenášaných dát. Transformáciu je možné vyžiada nastavením atribútu `transformation` na výstupnej štrukturovanej hodnote.

Downsampling

Downsampling je zredukovanie potu hodnôt poda daného kroku alebo na daný počet hodnôt. Použitý algoritmus sa snaží o najviac zachova priebeh krivky pôvodného asového radu a nevyhladzuje výslednú krivku ako bežné prevzorkovanie používajúce priemerovanie.

Vyžiadané zredukovanie algoritmom Largest Triangle Three Buckets na 100 hodnôt

```
{
  "type": "record",
  "structType": "SD.Arr_Real",
  "returnTransformation": {
    "type": "lttb",
    "threshold": 100
  }
}
```

Vyžiadané zredukovanie algoritmom Largest Triangle Three Buckets s krokom 86400 sekúnd (1 de)

```
{
  "type": "record",
  "structType": "SD.Arr_Real",
  "returnTransformation": {
    "type": "lttb",
    "step": 86400
  }
}
```

OHLC

OHLC (Open-High-Low-Close) algoritmus transformuje výstupný asový rad tak, že pre každý interval nájde prvú, maximálnu, minimálnu a poslednú hodnotu. Volitene pri tomto algoritme je možné nastavi i majú by intervaly spojité (po sebe nasledujúce intervaly majú spolonú poslednú a prvú hodnota - predvolene) alebo diskrétnu - atribút `discrete` a kde má by umiestnená asová znaka hodnoty reprezentujúcej celý interval (zaiatok "Start" alebo stred "Midpoint" - predvolene) - atribút `timestampPlacement`.

Vyžiadané OHLC transformácie s krokom 86400 sekúnd (1 de) s diskretnými intervalmi a asovými znakami na zaiatku intervalu

```
{
  "type": "record",
  "structType": "SD.Arr_Real",
  "returnTransformation": {
    "type": "ohlc",
    "step": 86400,
    "discrete": true,
    "timestampPlacement": "Start"
  }
}
```